CLAIMS

What is claimed is:

1. A system for automatically maintaining at least one deployment descriptor, comprising:

a parser capable of generating a representation of the at least one deployment descriptor;

a generator capable of creating the at least one deployment descriptor;

a validator capable of validating the at least one deployment descriptor;

wherein the system is capable of automatically repairing a first deployment descriptor of the at least one deployment descriptors if the first deployment descriptor is defective; and

wherein the system is capable of automatically deploying an application associated with the at least one deployment descriptor.

2. The system of claim 1 wherein:

the validator is capable of generating an error when it encounters a syntactic or semantic fault in the at least one deployment descriptor.

3. The system of claim 2 wherein:

in response to a user action, the IDE can navigate a user interface to the source of the error.

4. The system of claim 1 wherein:

the generator is capable of producing the at least one deployment descriptor from at least one source code file.

5. The system of claim 1, further comprising:

a builder component capable of automatically updating the at least one deployment descriptor to reflect one or more changes in at least one source code file.

6. The system of claim 1 wherein:

the representation can include information pertaining to at least one of: a Java™ archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a

Java™ Connector Architecture Component (RAR).

7. The system of claim 1 wherein:

the at least one deployment descriptor can be expressed as an Extensible Markup Language document.

8. A system for automatically maintaining at least one deployment descriptor, comprising:

a parser capable of generating a first representation of the at least one deployment descriptor;

a generator capable of creating a second representation of at least one deployment descriptor based on one or more source files;

a builder capable of comparing the first representation with the second representation;

wherein the builder is capable of updating the first representation based on the second representation if the first representation is out-of-date; and

wherein the system is capable of automatically repairing a first deployment descriptor of the at least one deployment descriptors if the first deployment descriptor is defective.

9. The system of claim 8 wherein:

the generator is capable of producing the at least one deployment descriptor from at least one source code file.

10. The system of claim 8 wherein:

a representation can include information pertaining to at least one of: a Java™ archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java™ Connector Architecture Component (RAR).

11. The system of claim 8 wherein:

the at least one deployment descriptor can be expressed as an Extensible Markup Language document.

12. The system of claim 8 wherein:

information is not deleted from the first representation.

13. The system of claim 8 wherein:

information in the second representation that is not in the first representation is added to the first representation.

14. The system of claim 8 wherein:

a user can modify information in the second representation via the IDE.

15. A method for updating at least one deployment descriptor, comprising:

creating a first representation of the at least one deployment descriptor;

creating a second representation of a second at least one deployment descriptor based on one or more source files;

comparing the first representation with the second representation; and

updating the first representation based on the second representation if the first representation is out-of-date.

16. The method of claim 15 wherein:

the at least one deployment descriptor can include information pertaining to at least one of: a Java™ archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java™ Connector Architecture Component (RAR).

17. The method of claim 15 wherein:

the at least one deployment descriptor can be expressed as an Extensible Markup Language document.

18. The method of claim 15 wherein:

information is not deleted from the first representation.

19. The method of claim 15 wherein:

information in the second representation that is not in the first representation is added to the first representation.

20. The method of claim 15 wherein:

a user can modify information in the second representation via the IDE.

21. The method of claim 15 wherein:

the IDE is capable of generating the second representation.

22. A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

create a first representation of the at least one deployment descriptor;

create a second representation of a second at least one deployment descriptor based on one or more source files;

compare the first representation with the second representation;

update the first representation based on the second representation if the first representation is out-of-date.

23. The machine readable medium of claim 22 wherein:

the at least one deployment descriptor can include information pertaining to at least one of: a Java™ archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java™ Connector Architecture Component (RAR).

24. The machine readable medium of claim 22 wherein:

the at least one deployment descriptor can be expressed as an Extensible Markup Language document.

25. The machine readable medium of claim 22 wherein:

information is not deleted from the first representation.

26. The machine readable medium of claim 22 wherein:

information in the second representation that is not in the first representation is added to the first representation.

27. The machine readable medium of claim 22 wherein:

a user can modify information in the second representation via the IDE.

28. The machine readable medium of claim 22 wherein:

the IDE is capable of generating the second representation.